

# The Role of Computational and Data Grids in Large-Scale Science and Engineering

William E. Johnston<sup>a</sup>

*Lawrence Berkeley National Laboratory and NASA Ames Research Center*

## Keywords

Grids; heterogeneous, widely distributed computing; NASA's Information Power Grid (IPG); Grid applications

## Abstract

As the practice of science moves beyond the single investigator due to the complexity of the problems that now dominate science, large collaborative and multi-institutional teams are needed to address these problems.

In order to support this shift in science, the computing and data handling infrastructure that is essential to most of modern science must also change in order to support this increased complexity. This is the goal of computing and data Grids: Software infrastructure that facilitates solving large-scale problems by providing the mechanisms to access, aggregate, and manage the computer network based infrastructure of science. This infrastructure includes computing systems, data archive systems, scientific instruments, and computer mediated human collaborations.

This paper examines several large-scale science problems, their requirements for computing and data Grid infrastructure, and the current approaches to providing the necessary functionality.

---

<sup>a</sup> [wejohnston@lbl.gov](mailto:wejohnston@lbl.gov), [www.itg.lbl.gov/~wej](http://www.itg.lbl.gov/~wej), [wej@nas.nasa.gov](mailto:wej@nas.nasa.gov), [www.ipg.nasa.gov](http://www.ipg.nasa.gov)

# 1 Introduction

---

“Grids” (see [1]) are an approach for building dynamically constructed problem solving environments using geographically and organizationally dispersed high performance computing and data handling resources.

Functionally, Grids are tools, middleware, and services for

- o providing a uniform look and feel to a wide variety of distributed computing and data resources
- o supporting construction, management, and use of widely distributed application systems
- o facilitating human collaboration and remote access to, and operation of, scientific and engineering instrumentation systems
- o managing and securing this computing and data infrastructure

This is accomplished through a set of uniform software services (the Common Grid Services - described in more detail below) that manage and provide access to heterogeneous, distributed resources. These services may be summarized as:

|  |                                      |
|--|--------------------------------------|
| • information services for resource discovery                        | • resource specification and request |
| • resource co-scheduling   | • uniform data access                |
| • authentication and authorization                                   | • security services                  |
| • auditing   | • monitoring                         |
| • global event services  | • global queuing                     |
| • data cataloguing, publishing, and subscribing                      | • resource brokering                 |
| • collaboration and remote instrument management and access services | • data location management           |
| • communication services   | • fault management                   |

The overall motivation for the current large-scale (multi-institutional) Grid projects is to enable the resource interactions that facilitate large-scale science and engineering such as aerospace systems design, high energy physics data analysis, climatology, large-scale remote instrument operation, etc.

The vision for computing, data, and instrument Grids is that they will provide significant new capabilities to scientists and engineers by facilitating routine construction of information based problem solving environments that are built on-demand from large pools of resources. That is, Grids will routinely – and easily, from the user’s point of view – facilitate applications such as:

- o coupled, multidisciplinary simulations too large for single computing systems (e.g., multi-component turbomachine simulation – see [2] and [3])
- o management of very large parameter space studies where thousands of low fidelity simulations explore, e.g., the aerodynamics of the next generation space shuttle in its many operating regimes (from Mach 27 at entry into the atmosphere to landing)
- o use of widely distributed, federated data archives (e.g., simultaneous access to metrological, topological, aircraft performance, and flight path scheduling databases supporting a National Air Transportation Simulation system)

- o coupling large-scale computing and data systems to scientific and engineering instruments so that complex real-time data analysis results can be used by the experimentalist in ways that allow direct interaction with the experiment (e.g. Cosmology data analysis involving telescope and satellite interaction, and coupling to simulations)
- o single computational problems too large for any single system (e.g. extremely high resolution rotocraft aerodynamic calculations)

The remainder of this paper is organized as follows:

- 1 Introduction**
- 2 Motivating Applications**
- 3 Application Needs**
- 4 A Model for Computing and Data Grids**
- 5 Current State of Grids**
- 6 Application Use of Grids**
- 7 Future Directions**
- 8 Acknowledgements**
- 9 Notes and references**

## **2 Motivating Applications**

---

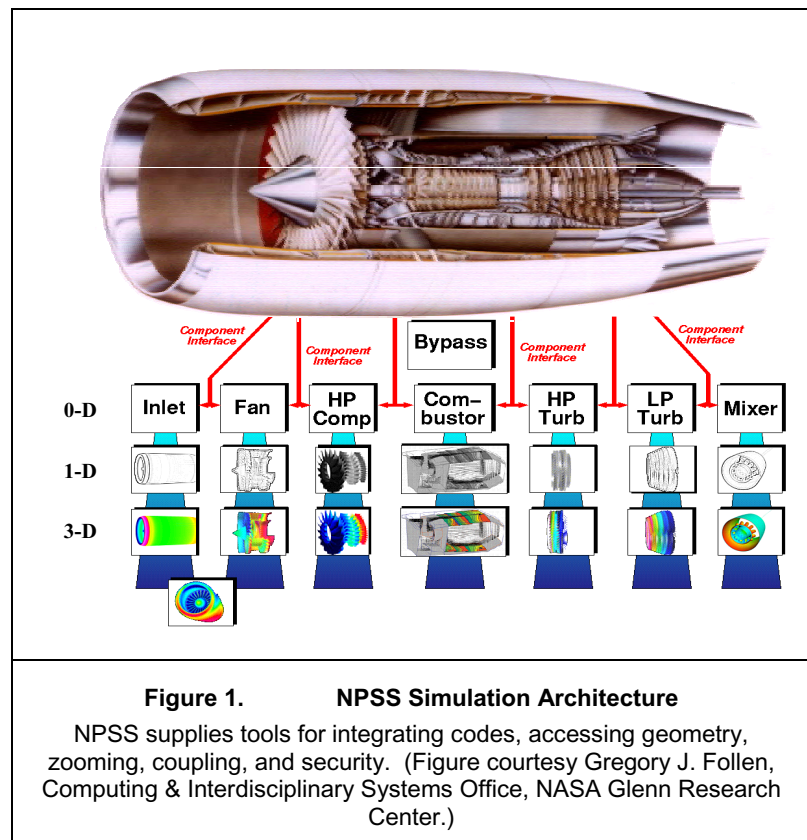
As the problem tackled by the science and research engineering communities become more and more complex, the computing requirements are not just for more computing power, but for dealing with more complex application systems as well. The several different examples presented below require both very high capability computing and data handling, and also require a complex mix of resources – multiple computers, databases, archives, instruments, etc., all of which must be carefully coordinated to solve the problem.

## 2.1 NASA's Aviation Safety Program

A current NASA R&D<sup>a</sup> project is to develop the approach and technology for modeling the entire commercial airspace of the US; that is, to produce a virtual national air space. The benefits of this range from potentially much more efficient utilization of airports and flight paths, to determining, and possibly correcting, aircraft related emergency conditions while in flight. The modeling involves integrating huge amounts of flight and ground operations data, weather, terrain, etc., together with whole aircraft simulations of the approximately 22,000 commercial flights per day in the US.

Most of the aircraft sub-systems (engine, wing lift, control surfaces, landing gear, etc.) are well studied individually, but combining these into a whole system simulation of the aircraft, and then integrating the result into an operational air space, is a considerable challenge.

The NPSS program [2] at NASA Glenn is working on coupling the many component models required to simulate an operational jet engine (see Figure 1), and integrating the resulting engine model with operational data. The sub-system simulations have been developed over a long time and they are written in a variety of languages (e.g. FORTRAN) and in a variety of styles. The NPSS program has built an application framework for coupling these together [3], and this approach is being extended to the whole aircraft problem.

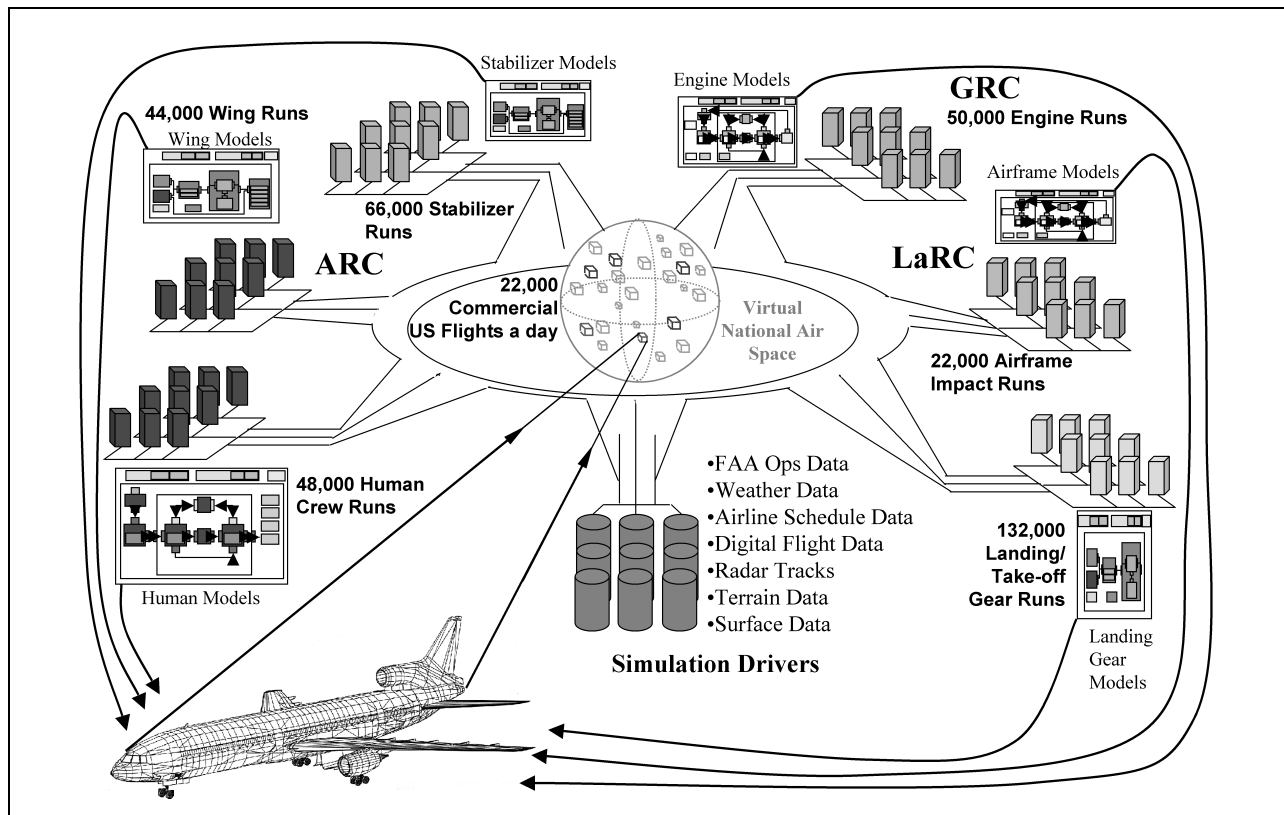


**Figure 1. NPSS Simulation Architecture**

NPSS supplies tools for integrating codes, accessing geometry, zooming, coupling, and security. (Figure courtesy Gregory J. Follen, Computing & Interdisciplinary Systems Office, NASA Glenn Research Center.)

<sup>a</sup> The “Virtual National Air Space” is the vision of Yuri Gawdiak and Bill McDermott, NASA Ames, and John Lytle and Gregory Follen, NASA Glenn.

Inserting the combined simulations into an operational environment means that the model drivers – initial conditions, boundary conditions, forcing functions, etc., – are now derived from various environmental data – observed velocity, elevation, pilot’s throttle setting, etc. This requires changing from static input files to potentially dynamic databases or live data feeds, and perhaps



**Figure 2. National Air Space Simulation Environment**

Many aircraft, flight paths, airport operations, and the environment are combined to get a virtual national airspace. (Figure courtesy Yuri Gawdiak, NASA Ames.)

having to build secondary models that convert observed quantities into the quantities needed to drive, e.g., the engine simulations.

Figure 2 illustrates the end-game in which all of the aircraft are simulated in the National air space, and these simulations are combined with the operational data to produce the Virtual National Air Space<sup>a</sup>. Such a whole systems simulation will clearly involve managing and coordinating a large number of organizationally and geographically dispersed computing and data resources: The computing power, data archives, special databases maintained by discipline experts, etc., will never be in one location or institution.

## 2.2 DOE’s Supernova Cosmology Program

Over the past several years, astronomers and astrophysicists have been conducting in-depth sky searches with the goal of identifying certain reference types of supernovae in their earliest evolutionary stages and then, during the two to four weeks of their most “explosive” activity, measuring their changing magnitude and spectra. These “standard candles,” as they are called by

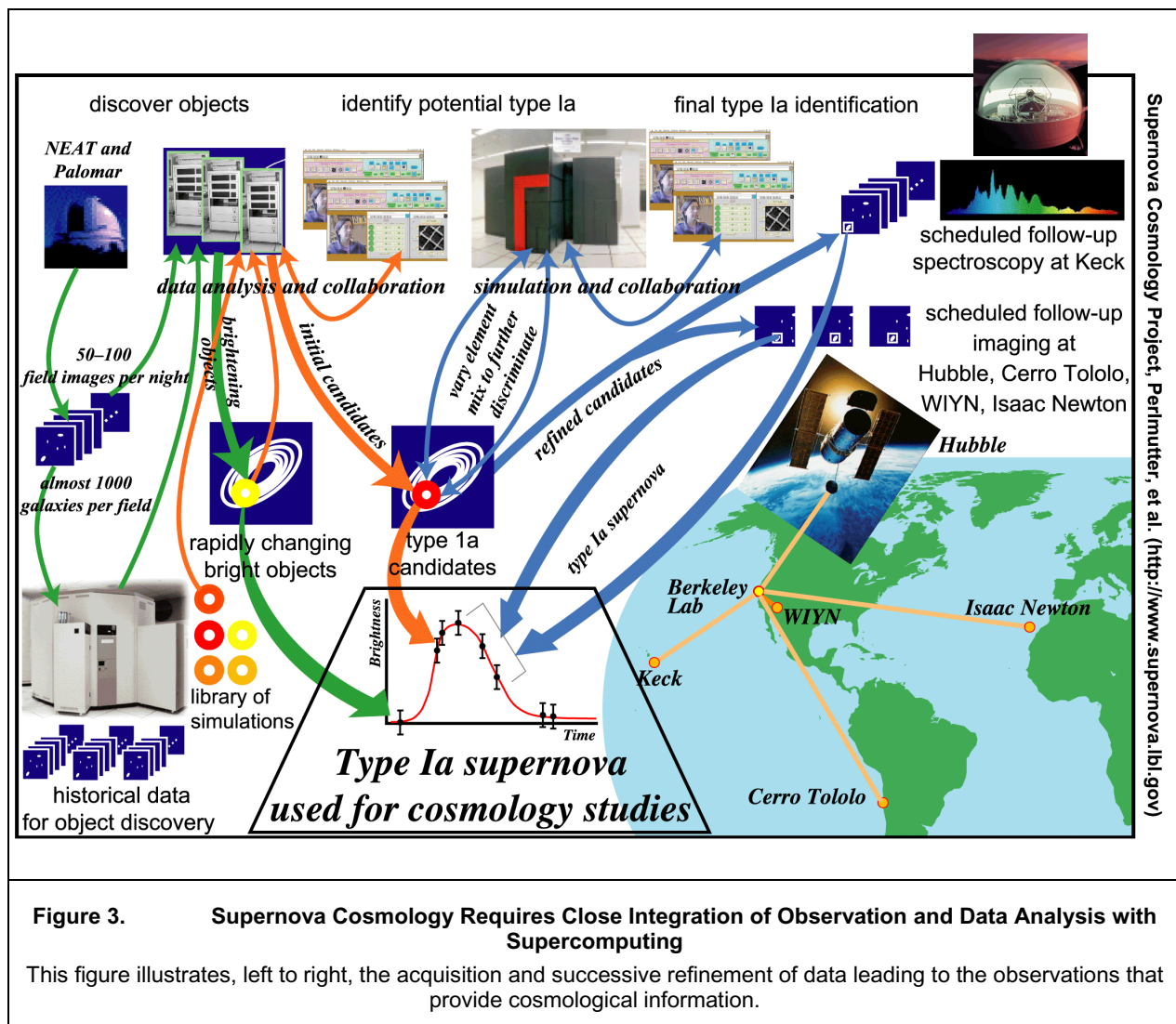
<sup>a</sup> This scenario does not imply complete, high fidelity simulation of every aircraft in the air space. Most of the simulations are likely to be run as low fidelity simulations that can rapidly be converted to high fidelity simulations, if the need arises.

the astronomers, are supernova that can be used to directly measure various cosmological properties. (See [4] and [5].) These early experiments have demonstrated that the expansion of the universe is accelerating, apparently driven by an unknown new force that overwhelms the force of gravity, contrary to existing models where gravity would cause the universe expansion to slow. The discovery of this new force – now called dark energy – is a stunning discovery and was named the “breakthrough of the year” by Science Magazine in Dec.1998.

These experiments have been daunting tasks in terms of both the number and volume of observations required. The early successes have driven the expansion of these searches in terms of both sky area and apparent magnitude observed. The search program currently under development at LBNL, the Supernova Factory (<http://snfactory.lbl.gov>), is an earth-based observation program utilizing observational instruments at Haleakala and Mauna Kea, Hawaii and Mt. Polomar, California. When fully implemented, this search program will also utilize instruments at observatories in Chile and the Canary Islands. This program will also serve as a development testbed for the next generation search program, the space-based Supernova Acceleration Probe (SNAP). The Supernova Acceleration Probe is a satellite-based supernova search program combining an optical field imager, near infrared imager and spectrometer in a single, dedicated spacecraft (see <http://snap.lbl.gov>).

This new approach to cosmology – only possible because of the availability of large-scale computing and data storage facilities at the DOE NERSC facility and the corresponding NSF supercomputer centers – is called “observational cosmology”.

The evolution from proof-of-principle to full scale supernova search has unveiled new operational issues for these research programs that we feel are characteristic of how modern science is evolving under the influence of vastly increased distributed computing and data handling capabilities. The first of these is the sheer scale of the computing and data-handling task involved. Raw, uncorrected sky images must be transferred nightly from remote observatories to central computing facilities, NERSC [6] in this case. Here, these images undergo extensive computational calibration and correction to eliminate sky tracking errors as well as instrumentation and atmospheric effects. The resulting images must then be compared to recent baseline sky catalogs in order to eliminate asteroids and man-made satellite tracks. Only then can automated search algorithms look for increases in stellar magnitude that may indicate the onset of supernova activity. Fifty plus Gigabytes in some 500 files need to be shepherded through this process of data transfer, computation and archiving on a daily basis for the 5 to 10 years duration of the search effort. The script and operator based automation used during early sky search programs simply will not scale to the levels of performance and reliability required by these new searches.



**Figure 3. Supernova Cosmology Requires Close Integration of Observation and Data Analysis with Supercomputing**

This figure illustrates, left to right, the acquisition and successive refinement of data leading to the observations that provide cosmological information.

Secondly, the amazing experimental results obtained thus far have promoted strong programmatic ties between cosmologists involved in modeling stellar behavior through simulations, and those engaged in direct observation. Simulation teams are now engaged in ambitious efforts to develop new models that provide full 3D simulation of both the hydrodynamic and radiative transfer aspects of supernova that can predict, based on the parameters of the exploding star, the spectra during supernova. Since the development of accurate models requires a detailed comparison with observed supernova data, data from the Supernova Factory is of critical importance in the successful development of these models. Although the initial motivation is the improvement of current computational models through direct and frequent comparison to observations, ultimately the goal is to use closely coupled observation/simulation efforts to filter out supernova candidates that are not the reference types useful for cosmology. As both the number of discovered supernovae and the demands for scarce, shared observational instruments increase, the ability to successfully filter unwanted supernovae out of the observational program becomes increasingly important. This is accomplished by using the initial observation to establish the parameters for the simulations, which, in turn, predict the observed spectra in order to determine the exact type of the supernova.

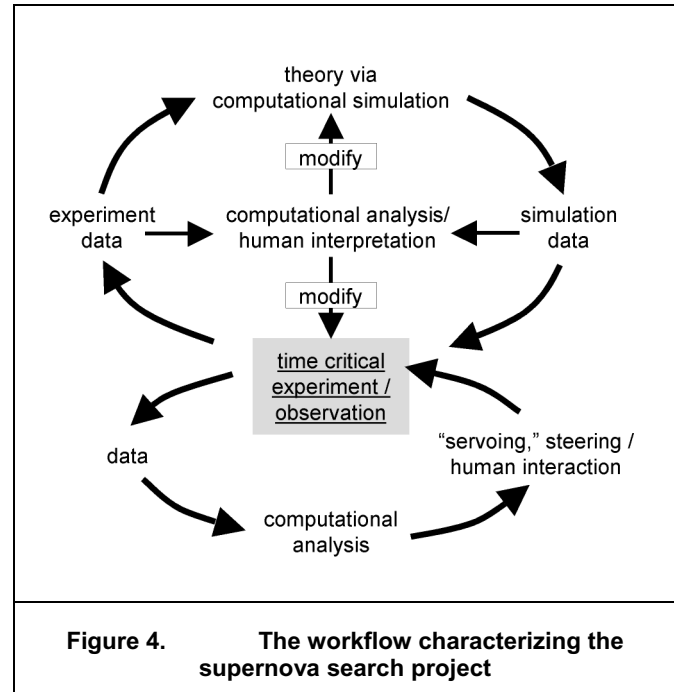
When this determination of type results in identifying a “standard candle” (type Ia) supernova, this information must be immediately conveyed to one of the large instruments such as Keck,



Palomar, or Hubble, in order to observe the spectra throughout the short (weeks long) life of the supernova. (See the right most process illustrated in Figure 3.) This is the information that permits cosmological inference.

The combination of these processes establishes a cycle of coarse observation – simulation – detailed spectra observation that is time constrained by the fact that the useful spectrum observation period is only for a few weeks following discovery. This cycle is illustrated in Figure 4.

Largely, this automation is required because of the sheer complexity of the operations involved. Input data and calibration files will need to be staged to disk before analysis programs can begin. Resulting data files will need to be archived, cataloged and published to other collaborating programs. In addition, in the presence software or hardware errors, these activities need to be rescheduled in a carefully controlled sequence to insure their proper completion. Some experiments, notably the Supernova Factory, will require that these operations occur, with a minimum of human intervention, around the clock, whenever observational data from earth and space instruments has been transferred to HPSS at NERSC. This workflow must be managed autonomously and reliably in order to meet the needs of the science.



**Figure 4. The workflow characterizing the supernova search project**

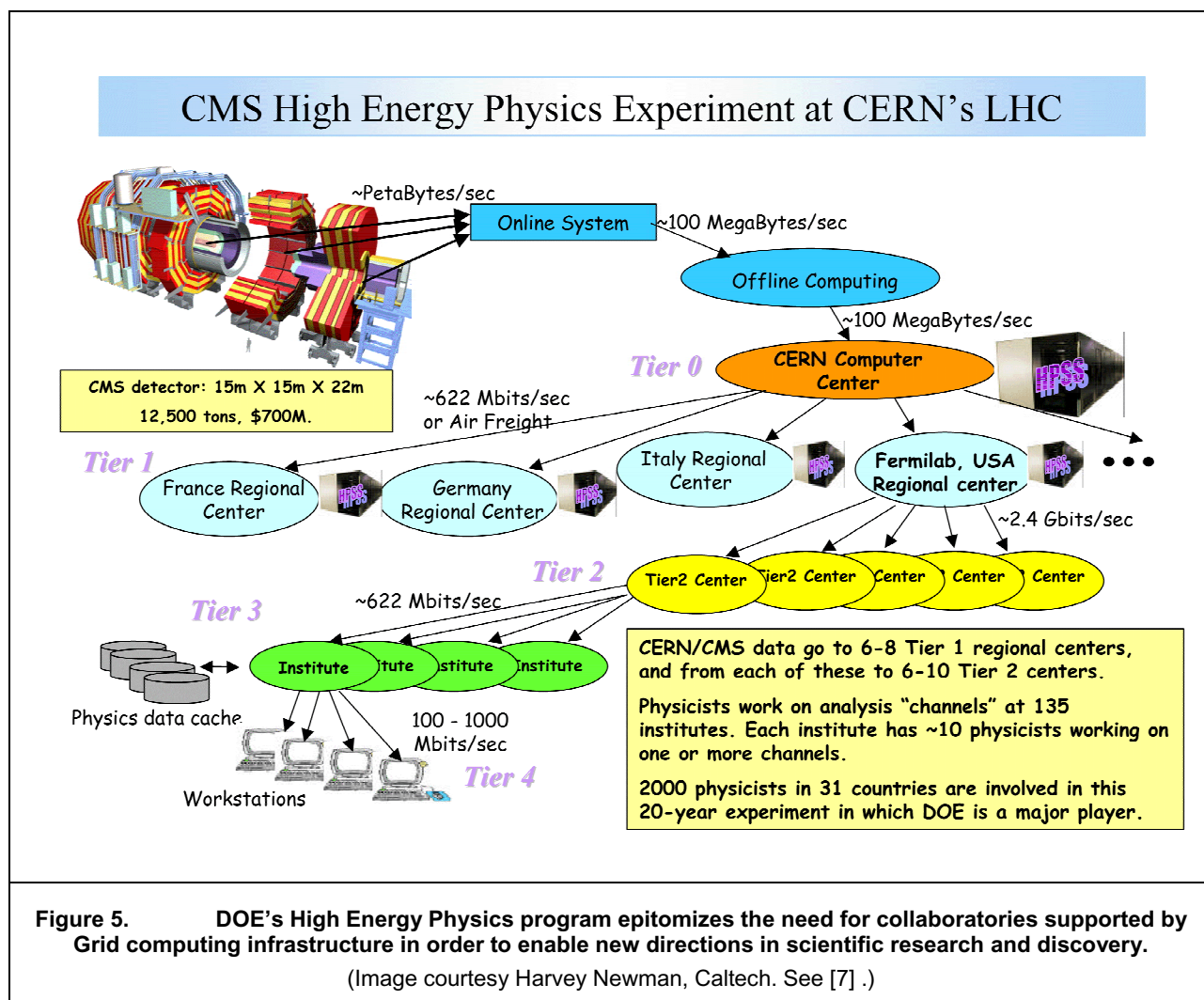
Finally, this work is inherently collaborative, and real-time collaboration is essential for its success. The scientists participating in the simulation development and in the measurements are themselves widely distributed, and furthermore, the telescopes, instruments and computers used in the search are distributed throughout the world. Effective interactions with staff at remote observatories will play an increasingly key role in successful daily operations. During both sky survey and follow up observations, it is often necessary to interact with observatory staff to adjust instrument settings, inquire about current sky conditions and quickly schedule repeat observations. Telephone and email are not effective for these interactions, and integrated collaboration tools become necessary.

Security and authorization also acquire significant importance when developing mechanisms that allow collaborators throughout the world to monitor and control daily analysis and archiving efforts. Success will depend on collaborating scientists being able to manage data processing and storage and to integrate advanced supernova simulation into the real-time control of the experiments. The ability to perform real-time control will allow collaborating scientists in one part of the world to look at results and change viewing plans in another part, thus taking advantage of the different time zones across the collaboration – an important aspect when observation can only be done for a few hours each night. This sort of access to the supercomputers and instruments must be able to be done securely or it will not happen.



## 2.3 High Energy and Nuclear Physics

One of the most basic science missions of the Department of Energy is the understanding of Matter and Energy in the Universe. The High Energy Physics program has as its primary focus the constituents of matter and the fundamental forces that govern their interactions. The Nuclear Physics program focuses on the structure of matter and the nuclear processes at work in the universe. The major particle and nuclear physics experiments of the next twenty years will break new ground in our understanding of the fundamental interactions and symmetries governing the



nature of matter and space-time [7]. The realization of these groundbreaking results involves the extraction of small or subtle new physics signals from large and potentially overwhelming backgrounds. Realizing the scientific wealth of these experiments presents new problems in data access, processing and distribution. Furthermore, these problems are being faced by ever growing collaborations of researchers spanning national and international networks, on a scale unprecedented in the history of science. There is a growing realization that, without the collaborative and workflow infrastructure that makes it possible for physicists in all world regions to contribute effectively to the analysis and the physics results, these research efforts will not succeed.

Further, the management and analysis of the extremely large quantities of data produced by leading high energy and nuclear physics experiments (e.g., BaBar, D0, RHIC, CMS, ATLAS)

represents an unprecedented information technology challenge. For these experiments to be successful, computing and data handling infrastructure must provide rapid, transparent access to experiment data samples and subsets drawn from massive datasets, growing from 100s of Terabytes in 2000 to Petabytes by 2005, and, ultimately, to 100 Petabytes (100 million megabytes) by 2010. There is a broad realization within these communities that the computational and storage resources needed for data management and analysis cannot realistically be gathered at a single location, and that future computational environments must hence be distributed collections of storage systems and compute farms, i.e. "Data Grids", that are operated in a coordinated fashion. E.g., see [8], [9], and [10].

These experiments collect specific types of data for the particles that result from high energy collisions of the protons, electrons, ions, etc. that are produced by the accelerators. The types of data are a function of the detector and include things like particle charge, mass, energy, 3D trajectory, etc. However much of science comes from inferring other aspects of the interactions by analyzing what can be observed. Many quantities are used in obtaining the scientific results of the experiment that are derived from what is observed. In doing this more abstract analysis, the physicist typically asks questions like<sup>a</sup>:

*Events of interest are usually characterized by a combination of jets of particles (coming from quark decays) and single particles like electrons and muons. In addition, we look for missing transverse energy (an apparent failure of momentum conservation) that would signal the presence of neutrinos that we cannot detect.*

*The topologies of individual events follow some statistical distributions so it is really the averages over many events that are of interest. In doing the analysis, we specify what cone angle would characterize a jet, how far one jet needs to be from another (in 3-dimensions), how far from the single particles, how much missing transverse energy, the angles between the missing energy vector and the other particles*

*What I would like to see is a set of tools to describe these topologies without typing in lots of code. A graphical interface that lets you draw the average event and trace out how statistical variations would affect that. We do simulation of interesting processes and they guide the selection of events, so we would want to learn from that.*

In order to transform these sorts of queries into combinations of existing tools and appropriate data queries, some sort of knowledge-based framework is needed.

## 2.4 Application Characteristics

---

From these examples, we can enumerate a set of high-level characteristics of the applications.

From the virtual sir space application, we can identify the following.

- o system simulations are built up by coupling legacy code components
- o computing capacity and simulation expertise will come from many different organizations
- o simulation components must be coordinated on many different computing systems
- o aircraft simulations must be coupled to the independent environmental and operations data sources that originate from hundreds of different locations

---

<sup>a</sup> Stewart Loken, Physics Division, Lawrence Berkeley National Laboratory

- o confidentiality of data and data access policy enforcement is required both for physical security of the aircraft and for protecting airline proprietary flight operations data
- o security and access control for the underlying computing and data archive systems must prevent service disruption

Characteristics of the supernova cosmology project include:

- o the cycle of coarse observation – simulation – fine observation represents a complex workflow that involves human interaction
- o the workflow process involves time constrained use of supercomputers
- o the final observations require interaction with potentially on-line instruments
- o the scientific teams are loose collaboration of researchers that are distributed world-wide
- o sufficient security and access control are required to prevent disruption and un-authorized access to the computing, data, and instrument systems

Characteristics of the HENP project include

- o massive data sets must be distributed world-wide for collaborative analysis
- o large-scale virtual organizations must be defined and managed
- o knowledge management systems that can map scientific queries to available tools and data

These application characteristics imply a range of computer science issues that are discussed in the next section.

### 3 Application Needs

---

The application characteristics from the examples above imply a collection of capabilities that must be addressed by the computer scientists who build the distributed systems that combine the computational tool with data and instruments..

In addition to the environment and services needed to support these applications, our experience in working with the design engineer / analyst who must use the system to accomplish a specific task suggests many other characteristics and requirements as well.. (E.g., see [11].)

In summary, these requirements include:

- Discipline analyst / problem solver requirements
  - o multiple datasets maintained by discipline experts at different sites that provide simulation geometry, performance data, and environmental conditions, must be accessed and updated by many collaborating analysts
  - o analysts must be able to securely share all aspects of their work process
  - o interfaces to data and computational tools must provide appropriate levels of abstraction for discipline problems solving
  - o management of very large datasets should not make the system unresponsive
  - o it should be easy to publish data to collaborators and easy to subscribe to data from collaborators

- o it should be easy to define, modify, and record descriptions of workflow
- o workflow definition should be possible via “visual programming” scenarios that integrate with the analyst “desktop” environment
- o collaborative, multi-party sharing of user interfaces, data, instruments, and computation should be provided
- Discipline tool builder requirements
  - o process and workflow management techniques – control of multi-step data analysis and simulations where software components and data will use computing and storage resources at different sites – must provide transparent and uniform control over all distributed resources participating in problem solving environments
  - o new approaches to computational simulation and data analysis must be accommodated in the distributed work/resource environment
  - o techniques are needed to describe and manage diverse strategies for parameter space exploration/filling
  - o mechanisms for managing generalized “faults” – data failures, program failures, both soft and hard, network partition, computing and data systems failures, etc. – are required for all aspects of the working environment
  - o location and architecture independent services must provide for various interprocess, interactive, data-intensive, and multi-point communication
  - o techniques are needed for debugging distributed software for correctness and performance
  - o it must be possible to audit and account for use of all resources
  - o co-allocation of resources to support coordinated use of multiple resources and scheduled use of resources must be available, and must accommodate “fuzzy” reservation (resource needed sometime in a given period)
  - o policy based quality-of-service should be available for all resources in order to support systems that have various “real-time” operating constraints
  - o CPU resource queuing mechanisms must provide a general and flexible control over all aspects of enqueued and running jobs
  - o global event management facilities must be able to signal job actions and application states
  - o use of CORBA, Java, Java/RMI, and DCOM must be provided within the context of the distributed resource environment
  - o generalized resource discovery services are needed in order to identify and characterize available resources
  - o support for remote execution management should include automatic selection and installation of code binaries and libraries appropriate for the target platform
  - o remote instrument interactions must be possible (Techniques are needed for coupling remote instrument system operation and data streams directly to computing and data management resources. Such systems should interoperate with tools supporting human sharing of computing environments.)

- o services to support collaboration

Operational environment requirements:

- o systems and operations professionals must be able to manage the distributed resources as part of a computing environment
- o resources should be “immune” to unauthorized access and manipulation
- o resource stakeholders/owners should have easily used mechanisms to enforce their use conditions and this must accommodate “fluid” work groups
- o the security and access control services must provide for easily specified characteristics and must be easily integrated into applications and problem solving environments

Many of these requirements are common to all of the science and engineering communities that will use Grids, and in the following, we only discuss additional requirements or different emphasis.

Requirements of the HENP community in addition to those above derive from a data analysis environment that is dominated by managing the location of data to be analyzed. Data must be moved around global networks and cached in, e.g., regional data centers, and from there to investigator data centers, etc. Requirements for this environment in addition to those above include:

- o data movement and use should be optimized so that it does not hinder the operation of the distributed application
- o mechanisms to facilitate community processing of data

### 3.1 The Role of Grids

---

The requirements of these several application areas lead to a characterization of the desired Grid functionality. This functionality may be represented as a hierarchically structured set of services and capabilities that are described below, and whose interrelationship is illustrated in Figure 6. Some of the key issues include:

- o techniques are needed for coupling heterogeneous computer codes, resources, and data sources in ways so that they can work on integrated/coupled problems in order to provide whole system simulations (“multi-disciplinary simulation/optimization”)
- o comprehensive network monitoring to locate, analyze, and correct bandwidth bottlenecks
- o data replica catalogues to provide global views of cached data
- o the methodology and implementation of incorporating, using, and managing resources in the overall environment must be scalable to thousands of resources

Scientific and Engineering applications involving distributed teams and distributed resources have lead to specific requirements for workflow and collaboration frameworks beyond those noted above. The basic workflow framework must provide for:

- o describing and managing multi-step, asynchronous component workflows, including managing fault detection and recovery
- o access to data and metadata publication and subscription mechanisms
- o event mechanisms - e.g. notification of when data or simulation results come into existence anywhere in the space of resources of interest

- o user interfaces to each of the above

Collaborative work support must address the human interaction aspects of collaborative data analysis:

- o maintenance of shared knowledge bases that allow a distributed community to create and update information about the state of overall progress of data processing, simulation results, existence of new, more highly refined, derived data, etc.
- o support for collaborative processing of data
- o support for on-line meetings, document sharing, and messaging
- o establishment and maintenance of the collaboration membership
- o security and management of access rights for the collaboration data and information

## 4 A Model for Computing and Data Grids

Grid environments that provide the services noted above are structured into a number of “levels”. (See Figure 6.) There are services that provide the user interfaces and application regime workflow management, tools and services supporting the development of application programs, the basic Grid services that provide uniformity and access to resources, and there are the resources themselves. Ancillary services such as security and system management are required at all levels.

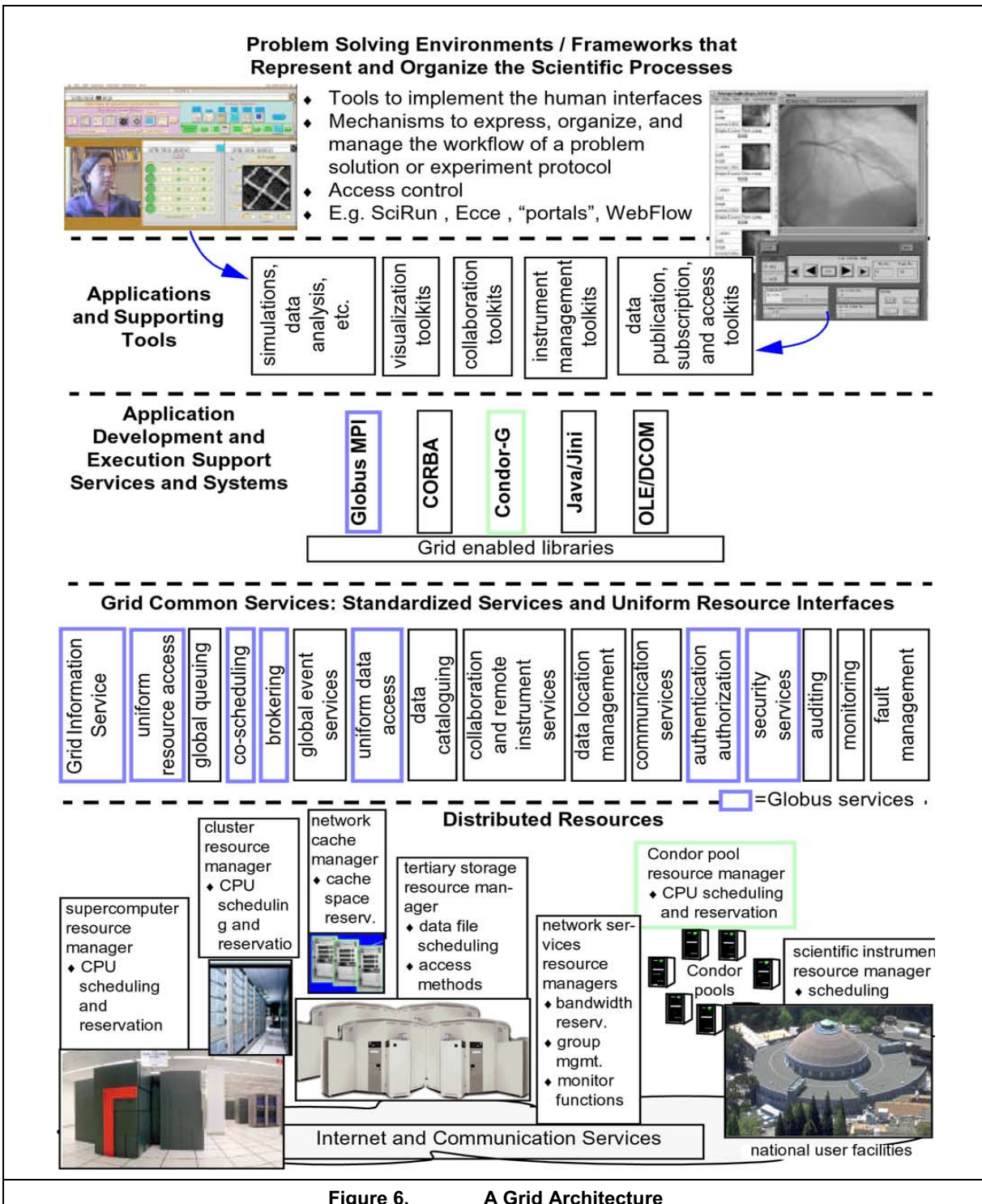


Figure 6. A Grid Architecture



An understanding is emerging as to what are the Grid services needed to support large-scale science and engineering activities. Many of the basic Grid services are currently available, and those are indicated in Figure 6. However, by no means are all of the required services currently available in Grids. Many of the more application oriented services like workflow and collaboration services are still under development.

#### **4.1 Problem Solving Environments: Knowledge Based Queries, User Interfaces and Workflow Management**

---

##### ***The Knowledge Grid<sup>a</sup>***

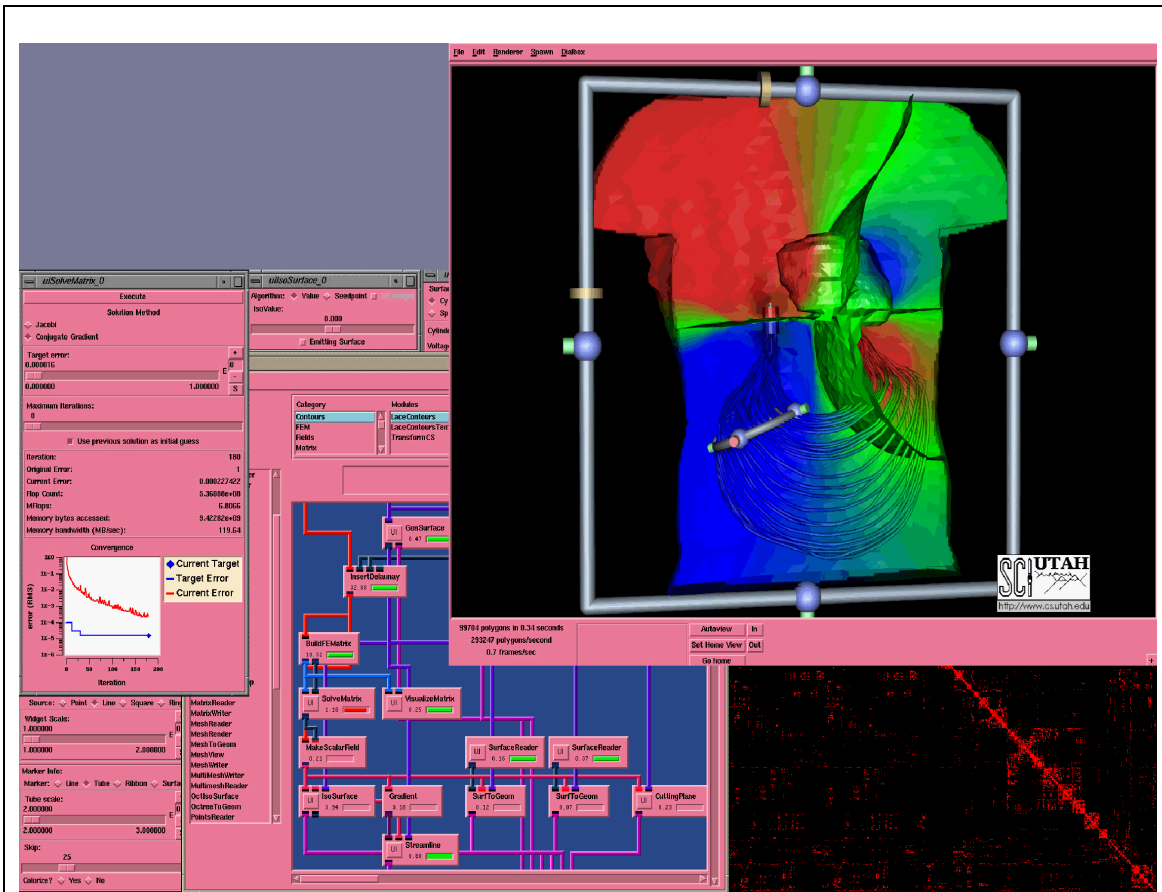
It is clear that for the Grid to realize the maximum impact on science and engineering that there must be mechanisms for discipline problem solvers to be able to express a problem in terms of the knowledge framework of their discipline, and then have that problem translated to the computational and data analysis operations of the underlying problem solving system. There have been various discipline specific efforts to do this sort of thing, but not much general infrastructure has been developed. The approach of Cannataro, et al [12] suggests one way to approach at least the representation and manipulation of the knowledge base that could translate moderately abstract queries into sets of computations and data analysis that resolve the query.

##### ***The User Interface: Integration with the Desktop***

A number of services directly support using the Grid by engineers or scientists. These include the toolkits for construction of application frameworks / problem solving environments (PSEs) that integrate Grid services and applications into the “desktop” environment. Services available in the user interface should include, for example, the graphical components (“widgets” / applets) for building application user interfaces; methods for control of the computer mediated; distributed human collaboration that support interface sharing and management; the tools that access the resource discovery and brokering services; the tools that provide generalized workflow management services such as resource scheduling and managing high throughput jobs, etc.

---

<sup>a</sup> I am indebted to Mario Cannataro, Domenico Talia, and Paolo Trunfio of the Istituto per la Sistemistica e l'Informatica (ISI) Consiglio Nazionale delle Ricerche (CNR), Italy, for this term.



**Figure 7. The SCIRun Problem Solving Environment**

SCIRun is a scientific programming environment that allows the interactive construction, debugging and steering of large-scale scientific computations. SCIRun can be used for interactively:

- o Changing 2D and 3D geometry models (meshes).
- o Controlling and changing numerical simulation methods and parameters.
- o Performing scalar and vector field visualization.

SCIRun uses a visual programming dataflow system. SCIRun is extensible to a variety of applications and will work with third party modules written in Fortran, C, and C++. (Image courtesy Prof. Chris Johnson, Univ. of Utah.)

All of these services should be available through Web / desktop interfaces in order to produce a highly usable environment. In this environment, problem-solving protocols may be formulated, controlled, modified, and integrated with other aspects of the work environment, and shared securely with collaborators. This sharing should be able to snapshot the current state of the PSE and pass this snapshot, or a functional replica of it, to a collaborator so that the same view of the application can be viewed and potentially manipulated by the collaborators. It should also be possible to use the PSE mechanism to insert “probes” into the workflow in order to monitor and diagnosis the functioning of the application system (as illustrated in Figure 7).

Systems like SciRun [13] (Figure 7), Ecce [14], and WebFlow [15], all provide elements of these, and are currently being converted to use Grid infrastructure for resource access and management. All of these also include some form of workflow management, but not as a general, standalone service.

## ***Workflow Management***

Reliable operation of large and complex data analysis and simulation tasks requires methods for their description and control. A workflow management system must provide for a rich and flexible description of the analysis processes and their inter-relationships, and also provide mechanisms for fault detection and recovery strategies in widely distributed systems.

Within the problem solving environments / frameworks must be mechanisms for describing the process of science: The protocols for experiments and hypothesis testing – the definition and management of the interplay of the data generation, data analysis, comparison with simulation, feedback to experiment, etc. Workflow management systems (e.g. as contained in the “framework” in the figure) will carry out the human defined protocols for, e.g., multi-disciplinary simulations and data analysis; global data cataloguing and replica management systems to manage the data for these scenarios, and global event services to manage the dynamic aspects of work protocols, will be essential adjuncts to the workflow engines. That is, these are the services needed directly by scientific and engineering problem solvers.

Then, through the use of appropriate distributed system services (i.e. the Grid Common Services described below), the workflow system will map these activities onto a sufficiently large and diverse set of computing and data handling resources (one of the goals of Grids is to provide such a pool) in order to not only accommodate the routine processing, but it have sufficient elasticity in the system to be able to rapidly locate and configure alternate resources in the event of faults.

## ***Collaboration tools***

Toolkits supporting the construction of PSEs must also provide the mechanisms for integrating computer mediated, distributed human collaboration into desktop problem solving environments. E.g. interface sharing, graphical user interface components that map to applications and Grid services, access control, a representation of the human work process that maps onto the workflow management mechanism, etc.

Collaboration tools must support the loosely bound collaborations of the scientific community (see [16]). Ideally these would include shared electronic notebooks, tele-meeting tools, tele-presence tools for laboratories and experiment sites, shared authoring tools, shared data publication tools. These tools will be built on Grid services such as secure group communication (“reliable multicast,” e.g., see [17]), which is the basic service for managing distributed, interacting, group services and the Grid Information Service for managing virtual organizations.

## 4.2 Programming Services

---

Tools and techniques are needed for building applications and applications systems that are built up from federated components that run in Grid environments. These techniques need to cover a wide spectrum of programming paradigms, and must operate in multi-platform, heterogeneous computing environments. For example, Grid enabled MPI [18] to support the IPC typical of numerical computations; Java and Python bindings to Grid services; CORBA integrated with Grid services to support building CORBA frameworks (like NPSS mentioned above), that federate application components; Condor-G [19] for managing large numbers of related jobs such as parameter studies and data analysis; Java/RMI and DCOM to obtain access to various commercial services; are all application oriented middleware systems that will have to interoperate with the Grid services in order to gain access to the resources managed by the Grid.

## 4.3 Grid Common Services

---

“Grid Common Services” refers to the basic services that provide uniform and location independent access and management of distributed resources. Much of the operational effort to run Grids is involved in maintaining these services.

Many Grids (including NASA’s IPG [20], DOE’s Science Grid [21], and ASCI Grid [22], and the Grids of the NSF Supercomputer centers[23]) currently use Globus [24] to provide the basic services that characterize and locate resources, initiate and monitor jobs, provide secure authentication of users, provide uniform access to data, etc.

### *Grid Information Service*

The Grid Information Service has the task of representing, and/or providing access to, virtually all aspects of the configuration and state of the Grid: resource characteristics, virtual organization scoping, persistent data catalogue locations, etc. It’s basic function is to be able to respond to queries about the availability of resources with certain characteristics and performance state, e.g. the computing systems architecture needed for a particular code. In large-scale environments it may also have the function of providing a rooted namespace into which can be inserted links to other directory services, such as definitions of virtual organizations, locations of data and data replica catalogues for discipline specific data, etc. The GIS must also provide for installing new objects/services into the Grid and must make these new objects known. In this role, the GIS also provides the mechanisms for defining the relationships among Grids, providing a framework for federating Grids, for administrative scoping, etc.

This service – currently provided by the Globus, Grid Information Service [25] – maintains detailed characteristics and state information about all resources, and will also need to maintain, or provide pointers to services that provide, dynamic performance information, information about current process state, user identities, allocations and accounting information.

### *Execution Management*

Several services are critical to managing the execution of application codes in the Grid. The first is resource discovery and brokering in order to build the (usually distributed and transient) platform – the ensemble of computing, data storage, etc., systems needed to support the application. By discovery we mean the ability to find the set of objects (e.g. databases, CPUs, functional servers) with a given set of properties that are needed by a distributed application system. Once the potential resources are identified by such queries to the GIS, the selection of the resources to actually be used is based on constraints such as allocation and scheduling is a

brokering function that will be built on the GIS services. The second is execution queue management, which relates to global views of CPU queues and their user-level management tools (Condor-G is an example an execution management tool). The third category is distributed application management. The last category includes tools for generalized fault management, for monitoring, and, e.g., supplying information to knowledge based recovery systems in the workflow management system.

### ***Runtime***

Runtime services include, e.g., checkpoint/restart mechanisms, access control, a global file system, and Grid communication libraries (such as a network-aware MPI) that support security, group communication (reliable multicast) and remote I/O.

Uniform naming and location transparent access must be provided for resources such as data objects, computations, instruments and networks. Transparent access requires uniform I/O mechanisms (e.g. read, write, seek) for all access protocols (e.g. http, ftp, nfs, Globus Access to Secondary Storage, etc.) and richer access and I/O mechanisms (e.g. “application level paging”) that are present in existing systems. Currently GridFTP [26] and MCAT/SRB [27] are providing some of these services.

High-speed, wide area, access to tertiary storage systems will always be critical for the science and engineering applications that we are addressing. High-performance applications require high-speed access to data files, and the Grid services must be able to stage, cache, and automatically manage the location of local, remote and cached copies of files. We are also going to need the ability to dynamically manage large, distributed “user-level” caches and “windows” on off-line data. Support for object-oriented data management systems will also be needed. Several of these services will become available over the next year or so from the GriPhyN (Grid Physics Network) project [28].

Services supporting collaboration and remote instrument control, such as secure, reliable group communication are needed. In addition, application monitoring and application characterization, prediction, and analysis, will be important for both users and the managers of the Grid. The NetLogger toolkit [29] and the Network Weather Service [30] are both being integrated as Grid services, and the GridForum, Grid Performance Working Group [31] is addressing this issue in a general way.

Monitoring services will need precision time event tagging for distributed, multi-component performance analysis. Generalized auditing of data file history and control flow tracking in distributed, multi-process simulations will be needed for integrity, change tracking, fault recovery, and security. General Grid event services are being addresses in the Grid Forum, Grid Computing Environments working group (see the GCE working group pages at [www.gridforum.org](http://www.gridforum.org) [32]).

## **4.4 Resource Management for Co-Scheduling and Reservation**

---

One of the most challenging and well known Grid problems is that of scheduling scarce resources such as supercomputers and large instruments. In many, if not most, cases the problem is really one of co-scheduling multiple resources. Any solution to this problem must have the agility to support transient applications based on systems that are built on-demand for limited periods of time, and in the case of Grid applications that analyze data from scientific and engineering experiment, the Grid resources are likely to have to be available on the sechedule of the instruments. In other words, not only will resources have to be co-secheduled, but they must

be scheduled for a particular time and date. CPU advance reservation scheduling and network bandwidth advance reservation are critical components to the co-scheduling services. In addition, tape marshaling in tertiary storage systems to support temporal reservations of tertiary storage system off line data and/or capacity is likely to be essential, and some of this is provided, e.g., by HRM [33]. The basic functionality for co-scheduling and/or resource reservation must almost always be provided by the individual resource managers, however Grid services serve to coordinate and provide uniform access to these resource specific services.

CPU advance scheduling services are currently provided, e.g., by PBSPro (<http://www.pbspro.com/>), and is a topic in the Grid Forum's Scheduling Working Group (see [32]).

#### **4.5 Access Control and Security**

---

The first requirement for establishing a workable authentication and security model for the Grid is to provide a single-sign-on authentication for all Grid resources based on cryptographic credentials that are maintained in the users desktop / PSE environment(s) or on one's person. This is provided by X.509 identity certificates or Kerberos credentials, together with the Globus proxies and the services that use them. See [34]. In addition, end-to-end encrypted communication channels are needed in for many applications in order to ensure data integrity and confidentiality.

The second requirement is an authorization and access control model that provides for management of stakeholder rights (use-conditions) and trusted third parties to attest to corresponding user attributes. A policy-based access control mechanism that is based on use-conditions and user attributes is also a requirement. Several approaches are being investigated for providing these capabilities (see, e.g., [35]) and work is being done on integrating these with Grids.

#### **4.6 Services for Operability: Operations and System Administration**

---

Implementing a persistent, managed Grid requires tools for deploying and managing the system software. In addition, tools for diagnostic analysis and distributed performance monitoring are required, as are accounting and auditing tools. Operational documentation and procedures are essential to managing the Grid as a robust production service.

To operate the Grid as a reliable, production environment is a challenging problem. Some of the identified issues include management tools for the Grid Information Service; diagnostic tools so operations/systems staff can investigate remote problems, and; tools and common interfaces for system and user administration, accounting, auditing and job tracking. Verification suites, benchmarks, regression analysis tools for performance, reliability, and system sensitivity testing are essential parts of standard maintenance.

Tools and documentation for operating production Grids are being developed at NCSA [36], and in the IPG [20] and DOE Science Grid [21] projects.

#### **4.7 The Architecture of Grids: How do all these services fit together?**

---

We conceptualize the Grid as a layered set of services, as illustrated in Figure 6, that manage the underlying resources, and middleware that supports different styles of usage (e.g. different programming paradigms and access methods).

However, the implementation is that of a continuum of somewhat hierarchically related, independent and interdependent services, each of which performs a specific function, and may rely on other Grid services to accomplish its function.

Further, the “layered” model should not obscure the fact that these “layers” are not just APIs and their underlying protocols, but usually a collection of functions and management systems that work in concert to provide the “service” at a given “layer.” The layering cannot be, and is not, rigid. “Drill down” (e.g. code written for specific system architectures and capabilities) must be easily managed by the Grid services.

Many of these services, and indeed the Grid architecture itself, are the subject of work in the Global Grid Forum, Grid Protocol Architecture Working Group [37]. Also see [38].

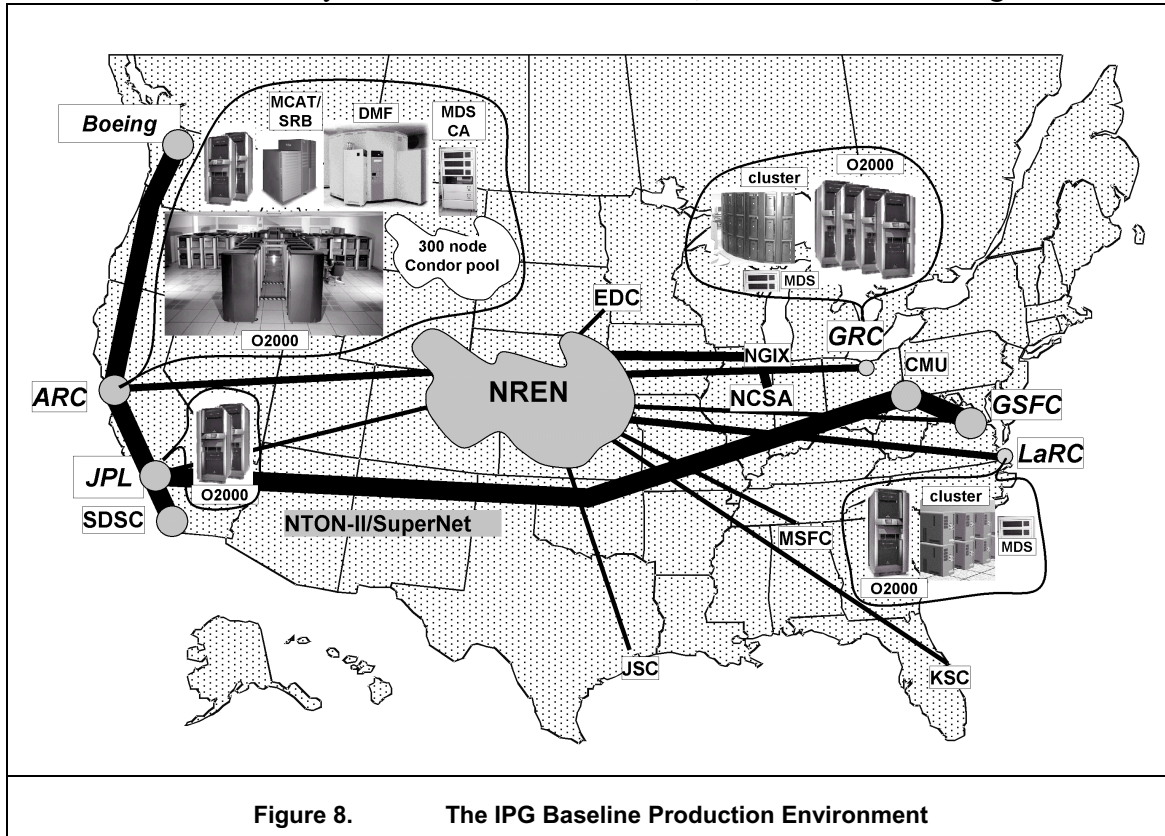
## 5 Current State of Grids

There are several Grids that are at, or close, to production status, and here we describe NASA’s Information Power Grid ([www.ipg.nasa.gov](http://www.ipg.nasa.gov)).

IPG has, over the past two and one half years, deployed a prototype production Grid. By production, we mean that the services and resources are persistent, there are operational groups responsible for those services and resources, and there is documentation and user support.

In the process of building this Grid environment a great deal is being learned about integrating Grids into production supercomputing environments, and some of the issues and lessons learned are documented in [39].

The current state of IPG may be characterized as follows, and is illustrated in Figure 8.





- Computing resources:
  - o approx. 1000 CPU nodes in half a dozen SGI Origin 2000s at NASA Ames, Glenn, Langley, and JPL
  - o 1024 node O2K and Cray SV-1 at Ames are almost ready to add to IPG (both are currently under test)
  - o several workstation clusters at Ames, Glenn, Langley, and JPL
  - o approx. 300 nodes in a Condor pool
- Wide area network interconnects of at least 100 Mbit/s
- Storage resources:
  - o 50-100 Terabytes of archival information/data storage uniformly and securely accessible from all IPG systems via MCAT/SRB and GSIFTP / GridFTP
- Globus providing the Grid Common Services

There are IPG operations groups in the NASA Ames Advanced Supercomputing (NAS) Division for:

- o Grid Information Services (the distributed master database of Grid resources)
- o Operation of the IPG computing and data systems at NAS, including the 1024 node O2K and SV-1
- o Globus software configuration and deployment
- o Grid security and authentication services, including the IPG X.509 Certification Authority
- o Grid enabled archival storage systems
- o User services
- o Condor workstation pool operation
- o The PBS batch scheduling system that provides advance reservation
- o Grid Accounting

In addition to the operational environment, IPG is providing and/or supporting research, development, and deployment work in numerous Grid technologies:

- o CORBA - Globus integration
- o Interoperation of Legion and Globus
- o CPU resource reservation
- o High throughput computing
- o Programming services
- o Distributed debugging
- o Grid enabled visualization
- o Parameter study frameworks
- o Network bandwidth reservation

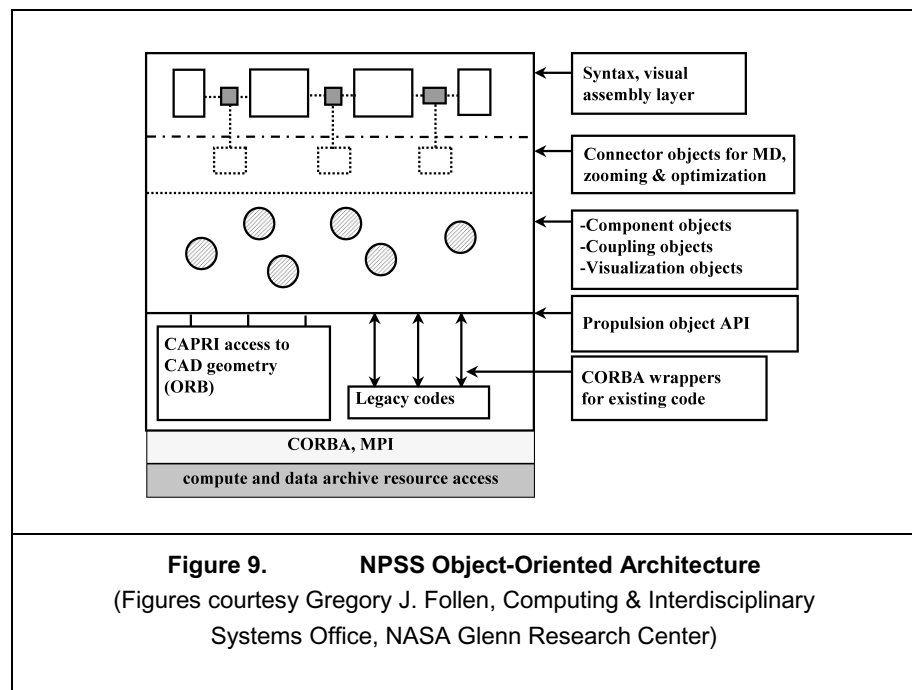
## 6 Application Use of Grids

---

In this section, we will cover some of the current applications use of Grids.

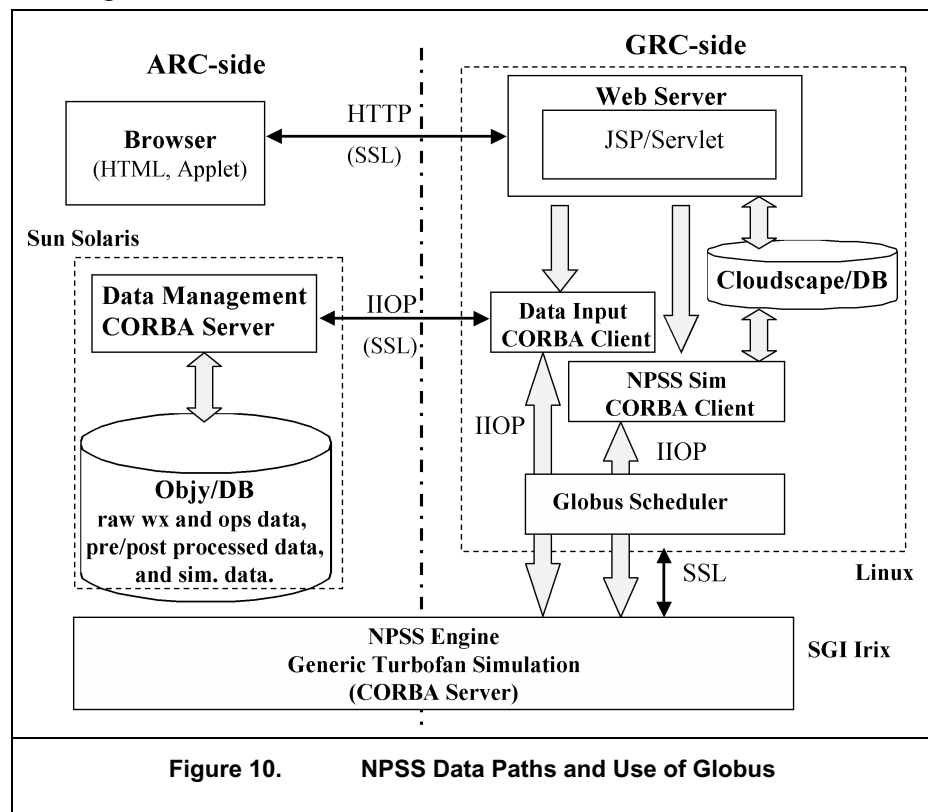
## 6.1 Aviation Safety Distributed Simulation

The NPSS system is a CORBA framework [3] in which the model components and the data are manipulated to solve various engine scenarios. The framework is indicated schematically in Figure 9. The framework data paths and use of Globus for instantiating and managing the CORBA environment on supercomputers is indicated in Figure 10.



## 6.2 Data Mining

The University of Alabama in Huntsville has developed a data mining system called ADaM (Algorithm Development and Mining). The current design consists of a mining engine and a daemon-controlled database. The database contains information about the data to be mined including its type and its location. To mine for data, the user provides the mining engine with a mining plan that consists of the sequential list of mining operations that are to be performed along with any parameters that may be required for each mining operation. The mining engine consults the database in order to find out where the data to be mined is stored and then applies the mining plan to the set of data that has been identified to the database. Each mining operation is represented as a shared-library file, one file per operation.



The IPG version of ADaM is structured so that the database and its associated daemon resides on a processor distinct from where the mining engine operates. For example, the database could be located on the user's workstation.

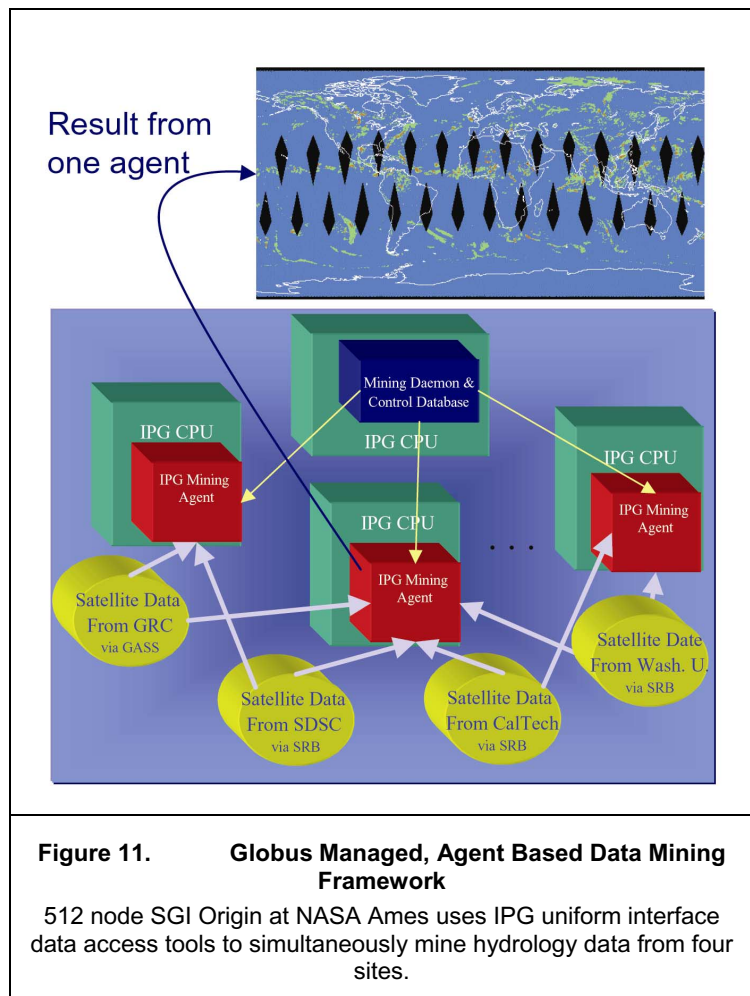
Using *globusrun*, the user is able to stage the mining engine to another system to execute. As required, the mining engine will acquire mining operation executables in the form of shared-library files from a mining operator repository on the IPG. Since a single mining plan may involve only a handful of operators (out of the 70+ operations that ADaM currently support), this means that only the required mining operators need to be sent to the IPG node that is currently supporting the mining engine. This is accomplished using the Globus data transfer functions (GASS) and MCAT/SRB.

As it executes, the mining engine stages the data to be mined from the data repository to the processor where the mining engine is executing. There are currently several sites that act as data repositories, and which currently pull data from NASA's Global Hydrology and Climate Center (which caches its most recent data holding in an FTP directory accessible through the web) so that it can be mined for severe storms.

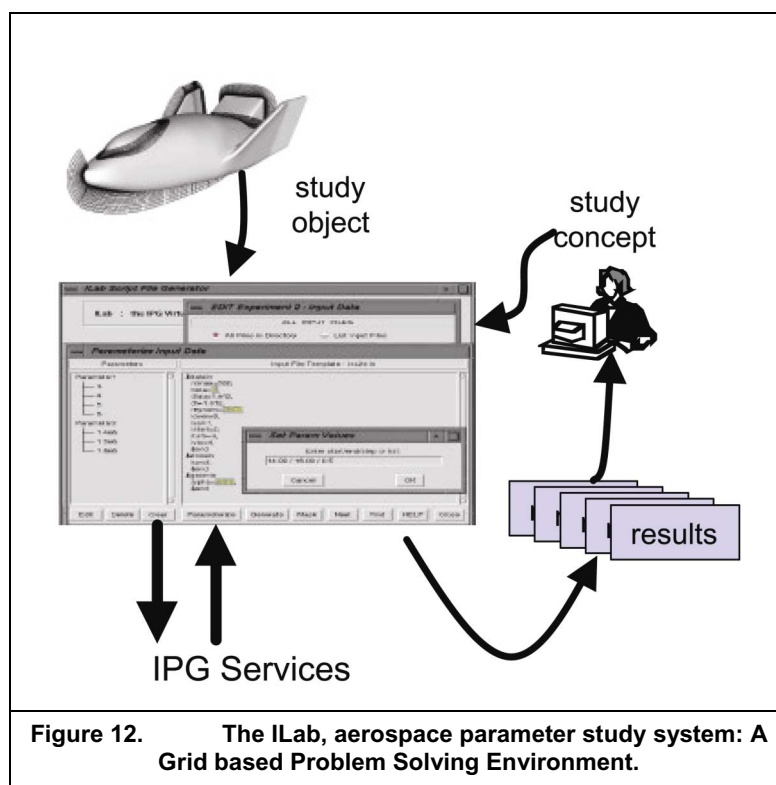
This is work of Tom Hinke (thinke@mail.arc.nasa.gov). See [40].

### 6.3 Parameter Studies and “Embarrassingly Parallel” Codes

ILab is an aerospace parameter study system that is designed to provide for substantial human efficiency in studying complex systems. It uses IPG to locate and manage compute resources for the individual jobs. See . This is work of Maurice Yarrow (yarrow@nas.nasa.gov), NASA Ames. See [41].



The IPG Condor pool (a computational “node” managed by Globus) is used for for certain types of jobs that are easily decomposed in small units. For example, a molecular design application coded in Java and managed by the Condor cycle scavenger is able to apply several gigaflop years of otherwise idle computing time to various problems in molecular design for nanotechnology devices and materials. These applications are coded in Java for platform independence, and the increased number of platforms where the code can run more than compensates for the computational inefficiency of Java. See [42] and [43]. This is work of Al Globus (globus@nas.nasa.gov), NASA Ames.



**Figure 12.** The ILab, aerospace parameter study system: A Grid based Problem Solving Environment.

## 7 Future Directions

Much of the current use of Grids falls into two categories. One is to instantiate a framework on the remote machine, and then proceed within that framework. The NPSS and data mining applications described above are examples of this. The second is to manage large numbers of small jobs (parameter studies and data analysis), and the ILAB systems described above is an example of this.

While these are early successes in using Grids, they are only a first step to the level of Grid technology and deployment that we need to have a substantial impact on large-scale science and engineering.

Another area that will be critical for Grids to facilitate applications with very large data handling problems such as the high-energy physics experiments described above. There are several “Data Grid” projects whose whole purpose is to address Grid technologies for this massive data handling, and the reader is referred to [28] which provides a very nice overview of the issues and approach.

Some early work on high latency numerical algorithms that will be required to distribute single applications across Grids is described in section 7.1.

Section 7.2 presents a list that represents, in the author’s opinion, some of the high priority items that need to be addressed in order for Grids to be able to routinely support widely distributed computing, data, and instrument applications for science and engineering. (The list is ordered roughly by the structure of the architecture diagram in Figure 6, not by priority.)

## 7.1 High Latency Algorithm R&D

---

For numerical computations (as opposed to modular application components) to be distributed across Grids will require new approaches that provide for tolerance of high and variable latency in intra algorithm communication. There is research using NASA's IPG for single simulations that operate across many, widely distributed systems. One candidate for CFD algorithms that accommodate the high and variable latencies encountered in Grid computing environments is overset grid codes that can tolerate time step mis-matches on the intra-object boundaries [44]. A version of the OVERFLOW, Navier-Stokes, CFD simulation code is being modified for this approach. It has been demonstrated operating across systems at ARC, GRC, and LaRC, solving for flow about large test objects mounted in a wind tunnel.(See [45].)

## 7.2 Grid R&D Areas

---

The general technology areas that need to be addressed and integrated into Grids in order to significantly increase the capabilities of Grids include the following.

- Knowledge Frameworks  
From problem description, identify appropriate computational components and (virtual) datasets
  - o Self describing components
  - o Self describing data
  - o Languages for representing “knowledge” – the characteristics and relationship of the semantics of application objects and operations
  - o Ontologies for describing and manipulating the conceptual structure of the application area
- Problem Solving Environments  
Mechanisms for representing and manipulating the structure of the computational representation of the problem
- Workflow management  
Provide for description and subsequent control of the ordered steps and events that represent a “job”. A general approach is needed to provide rule-based execution management system driven from published/subscribed global events (where the “events” represent process completion, file or other state creation, instrument turn-on, etc.).
  - o Generalized, global events with standard semantics, information carrying capability, various publish / subscribe mechanisms, and persistence
  - o Fault detection and recovery, including generalized faults such as data generated application mal-functions such as non-convergence of an iterative algorithm
- Collaboration frameworks  
Mechanisms for human control and sharing of all aspects of an executing workflow
  - o Probe insertion and management
  - o User and graphical interface replication and sharing
- Applications  
Algorithms and “wrapping” techniques that enable dynamic object management in an environment of widely distributed resources
  - o Latency tolerance
  - o Mobile work units

- Grid Common Services / Collective Management of the distributed environment
  - o Generalized resource discovery (anywhere, any time)
  - o Allocation negotiation
  - o Dynamic execution management
  - o Data replication / location management
  - o New approaches to organizing and managing the space of Grid objects (e.g. Web search engines operating on XML objects that represent all Grid resources, content addressable network services [46], Condor's Matchmaker [47], etc.)
- Security and Authorization
  - o Security gateways that map Grid security to the security mechanisms of DCOM and Java/RMI based servers
  - o Authorization mechanisms that accommodate policy involving multiple stakeholders providing use-conditions on resources and many different user attributes in order to satisfy the use-conditions
- Resources
 

Control and functionality of the physical resources: computing and data storage systems, networks, instruments

  - o Advance reservation
  - o Co-scheduling
  - o Deadline scheduling
  - o Strong and flexible access control
  - o Adaptive monitoring

## 8 Acknowledgements

---

Many people are working on Grids, and much of this work is represented by work areas in the Global Grid Forum ([www.gridforum.org](http://www.gridforum.org)). However, several people have contributed key ideas and insights over the past several years. Among these, the author has enjoyed discussions with Ian Foster, Carl Kesselman, Steve Tuecke, Dennis Gannon, Bill Nitzberg, Stewart Loken, and Bill Feiereisen. The DOE Science Grid is being built starting in the summer of 2001, and the principals in that project are the author, Ian Foster (Argonne National Lab.), Ray Bair (Pacific Northwest Lab), and Al Geist (Oak Ridge National Lab.). NASA's Information Power Grid project is managed collaboratively by the author, Arsi Vaziri, Leigh Ann Tanner, William Thigpen, Tom Hinke, and Piyush Mehrotra, all of NASA Ames Research Center.

IPG is funded primarily by NASA's Aero-Space Enterprise, Information Technology (IT) program (<http://www.nas.nasa.gov/IT/overview.html>). DOE's Science Grid is funded by the U.S. Dept. of Energy, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences Division (<http://www.sc.doe.gov/production/octr/mics>) under contract DE-AC03-76SF00098 with the University of California.

## 9 Notes and references

---

- [1] *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, eds. 1998, Morgan Kaufmann. [http://www.mkp.com/books\\_catalog/1-55860-475-8.asp](http://www.mkp.com/books_catalog/1-55860-475-8.asp)
- [2] "Numerical Propulsion System Simulation," NPSS. <http://hpcc.lerc.nasa.gov/npssintro.shtml>
- [3] "A CORBA-based Development Environment for Wrapping and Coupling Legacy Codes," G. Follen, C. Kim, I. Lopez, J. Sang and S. Townsend. In *Tenth IEEE International Symposium on High Performance Distributed Computing*. 2001. San Francisco.
- [4] "Supernova Cosmology Project," S. Perlmutter and e. al. 2001. <http://www.supernova.lbl.gov>
- [5] The "Cosmology Tutorial" is a good introduction to cosmology in general, and also specifically discusses supernova cosmology (at [http://www.astro.ucla.edu/~wright/sne\\_cosmology.html](http://www.astro.ucla.edu/~wright/sne_cosmology.html)) "Cosmology Tutorial," N. Wright. 2001, UCLA. <http://www.astro.ucla.edu/~wright/cosmolog.htm>
- [6] NERSC is one of the largest unclassified scientific supercomputer centers in the US. It's mission is to accelerate the pace of scientific discovery in the DOE Office of Science community by providing high-performance computing, information, and communications services. NERSC is the principal provider of high performance computing services to Office of Science programs - Magnetic Fusion Energy, High Energy and Nuclear Physics, Basic Energy Sciences, Biological and Environmental Research, and Advanced Scientific Computing Research. "National Energy Research Scientific Computing Center," NERSC. 2001. [www.nersc.gov](http://www.nersc.gov)
- [7] "Worldwide Distributed Analysis for the Next Generations of HENP Experiments," H. Newman. In *Computing in High Energy and Nuclear Physics*. 2000. Padova, Italy. [http://chep2000.pd.infn.it/abs/abs\\_e385.htm](http://chep2000.pd.infn.it/abs/abs_e385.htm)
- [8] "Particle Physics Data Grid", PPDG. 2000. <http://www.cacr.caltech.edu/ppdg/>
- [9] "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets", A. Chervenak, I. Foster, C. Kesselman, C. Salisbury and S. Tuecke. J. Network and Computer Applications, 2001.
- [10] The DataGrid Project is a proposal made to the European Commission for shared cost research and technological development funding. The project has six main partners:  
CERN - The European Organization for Nuclear Research near Geneva, Swiss;  
CNRS - France - Le Comité National de la Recherche Scientifique;  
ESRIN - the European Space Agency's Centre in Frascati (near Rome), Italy;  
INFN - Italy - Istituto Nazionale di Fisica Nucleare;  
NIKHEF - The Dutch National Institute for Nuclear Physics and High Energy Physics, Amsterdam, and;  
PPARC - United Kingdom - Particle Physics and Astronomy Research Council.



The objective of the project is to enable next generation scientific exploration which requires intensive computation and analysis of shared large-scale databases, from hundreds of TeraBytes to PetaBytes, across widely distributed scientific communities. We see these requirements emerging in many scientific disciplines, including physics, biology, and earth sciences. Such sharing is made complicated by the distributed nature of the resources to be used, the distributed nature of the communities, the size of the databases and the limited network bandwidth available. To address these problems we propose to build on emerging computational Grid technologies, such as that developed by the Globus Project EU\_DataGrid. 2001. <http://www.cern.ch/grid>

[11] "Information Power Grid Implementation Plan: Research, Development, and Testbeds for High Performance, Widely Distributed, Collaborative, Computing and Information Systems Supporting Science and Engineering," W. Johnston, D. Gannon and B. Nitzberg. 2001: NASA Ames. [http://www.ipg.nasa.gov/engineering/requirements/requirements\\_index.htm](http://www.ipg.nasa.gov/engineering/requirements/requirements_index.htm)

[12] "The Knowledge Grid: Towards an Architecture for Knowledge Discovery on the Grid," M. Cannataro, D. Talia and P. Trunfio. In *FIRST EUROGLOBUS WORKSHOP*. 2001. Robinson Club Apulia Village, Marina di Ugento, Lecce. <http://www.euroglobus.unile.it/>

[13] SCIRun is a scientific programming environment that allows the interactive construction, debugging and steering of large-scale scientific computations. "SciRun," C. Johnson. 2000. <http://www.cs.utah.edu/~sci/software/>

[14] "Open Data Management Solutions for Problem Solving Environments: Application of Distributed Authoring and Versioning (DAV) to the Extensible Computational Chemistry Environment," K. Schuchardt, J. Myers and E. Stephan. In *High Performance Distributed Computing - 10*. 2001. San Francisco, CA. [http://www.emsl.pnl.gov:2080/docs/collab/about\\_us/Resources.html](http://www.emsl.pnl.gov:2080/docs/collab/about_us/Resources.html)

[15] "WebFlow - High-Level Programming Environment and Visual Authoring Toolkit for High Performance Distributed Computing," G. Fox. In *Supercomputing*. 1998. [http://www.supercomp.org/sc98/TechPapers/sc98\\_FullAbstracts/Akarsu809/Index.htm](http://www.supercomp.org/sc98/TechPapers/sc98_FullAbstracts/Akarsu809/Index.htm)

[16] The PCCE is a web-based persistent space that supports continuous and ad hoc collaboration. It targets daily tasks such as document sharing and provides a base connectivity for communication. Documents can be published into the web environment (by users or applications) and viewed in a browser via WebDAV capabilities. "Pervasive Collaborative Computing Environment," D. Agarwal. <http://www-itg.lbl.gov/Collaboratories/pcce.html>

[17] "Reliable and Secure Group Communication," D. Agarwal, K. Berket and O. Chevassut. 2001. <http://www-itg.lbl.gov/CIF/GroupComm>

[18] "A Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems," I. Foster and N. Karonis. In *SC98*. 1998. <http://www.globus.org/research/papers.html#mpich98>

[19] "Condor-G," J. Frey, T. Tannenbaum, M. Livny, I. Foster and S. Tuecke. In *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*. 2001: IEEE Press. <http://www.globus.org/research/papers.html#Condor-G-HPDC>

[20] "NASA's Information Power Grid," IPG. <http://www.ipg.nasa.gov>

[21] The DOE Science Grid's major objective is to provide the advanced distributed computing infrastructure based on Grid middleware and tools to enable the degree of scalability

in scientific computing necessary for DOE to accomplish its missions in science. "DOE Science Grid," Science\_Grid. 2001. <http://www-itg.lbl.gov/Grid/>

[22] "Constructing the ASCI Grid," J. Beiriger, W. Johnson, H. Bivens, S. Humphreys and R. Rhea. In *Proc. 9th IEEE Symposium on High Performance Distributed Computing*. 2000: IEEE Press.

[23] The NSF PACIs are the Alliance/NCSA (<http://www.ncsa.uiuc.edu/>) and NPACI/SDSC (<http://www.npaci.edu/>) "PACI," PACI.

[24] "Globus: A Metacomputing Infrastructure Toolkit", I. Foster and C. Kesselman. Int'l J. Supercomputing Applications, 1997. <http://www.globus.org/research/papers.html>

[25] "Grid Information Services for Distributed Resource Sharing," K. Czajkowski, S. Fitzgerald, I. Foster and C. Kesselman. In *High Performance Distributed Computing - 10*. 2001. San Francisco, CA. <http://www.globus.org/research/papers.html#MDS-HPDC>

[26] "GridFTP: Universal Data Transfer for the Grid", Globus Project. 2001. <http://www.globus.org/datagrid/>

[27] "Massive Data Analysis Systems," R. Moore and et al., San Diego Supercomputer Center. <http://www.sdsc.edu/MDAS>

[28] The GriPhyN (Grid Physics Network) collaboration is a team of experimental physicists and information technology (IT) researchers who plan to implement the first Petabyte-scale computational environments for data intensive science in the 21st century. Driving the project are unprecedented requirements for geographically dispersed extraction of complex scientific information from very large collections of measured data. To meet these requirements, which arise initially from the four physics experiments involved in this project but will also be fundamental to science and commerce in the 21st century, GriPhyN will deploy computational environments called Petascale Virtual Data Grids (PVDGs) that meet the data-intensive computational needs of a diverse community of thousands of scientists spread across the globe. "GriPhyN (Grid Physics Network) project proposal," GriPhyN. 2000. <http://www.griphyn.org/info/itr2000.html>

[29] "NetLogger: A Toolkit for Distributed System Performance Analysis," D. Gunter, B. Tierney, B. Crowley, M. Holding and J. Lee. In *IEEE Mascots 2000: Eighth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. 2000. <http://www-didc.lbl.gov/papers/NetLogger.Mascots.paper.ieee.pdf>

[30] "Forecasting Network Performance to Support Dynamic Scheduling Using the Network Weather Service," R. Wolski, in *Proc. 6th IEEE Symp. on High Performance Distributed Computing*. 1997: Portland, Oregon.

[31] The Grid Performance working group is focused on defining standards and best practices for the gathering, representation, storage, distribution, and query of performance information about Grid resources and applications. R. Aydt, D. DGunter and W. Smith. 2001. <http://www-didc.lbl.gov/GridPerf/>

[32] The Global Grid Forum ([www.Gridforum.org](http://www.Gridforum.org)) is an informal consortium of institutions and individuals working on wide area computing and computational Grids: the technologies that underlie such activities as the NCSA Alliance's National Technology Grid, NPACI's Metasystems efforts, NASA's Information Power Grid, DOE ASCI's DISCOM program, and other activities worldwide. Grid\_Forum.

- [33] "Storage Management for High Energy Physics Applications," A. Shoshani, L. M. Bernardo, H. Nordberg, D. Rotem and A. Sim. In *Computing in High Energy Physics 1998 (CHEP 98)*. 1998. <http://gizmo.lbl.gov/~arie/download.papers.html>
- [34] "A Security Architecture for Computational Grids," I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, in *ACM Conference on Computers and Security*. 1998. p. 83-91.
- [35] "Certificate-based Access Control for Widely Distributed Resources," M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson and A. Essiari. In *Eighth Usenix Security Symposium*. 1999. <http://www-itg.lbl.gov/Akenti/papers.html>
- [36] "Alliance Computational Environments & Security," NCSA. 2001. <http://www.ncsa.uiuc.edu/TechFocus/Projects/>
- [37] The role of the Grid Protocol Architecture Working Group is to provide a conceptual framework for discussing the interrelationships, completeness, and minimality of the protocol approach to Grid services that is coming out of GF. "Global Grid Forum, Grid Protocol Architecture Working Group," W. Johnston. 2001. <http://www-itg.lbl.gov/GPA>
- [38] "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", I. Foster, C. Kesselman and S. Tuecke. Intl. J. Supercomputer Applications, 2001. **(to appear)**. <http://www.globus.org/research/papers/anatomy.pdf>
- [39] "NASA's Information Power Grid: Production Grid Experience with Distributed Computing and Data Management," W. Johnston. In *Second Global Grid Forum Workshop (GGF2)*. 2001. Washington, D.C. <http://www.itg.lbl.gov/~johnston/Grids>
- [40] "Data Mining on NASA's Information Power Grid," T. Hinke and J. Novonty. In *Ninth IEEE International Symposium on High Performance Distributed Computing*. 2000.
- [41] "An Advanced User Interface Approach for Complex Parameter Study Process Specification on the Information Power Grid," M. Yarrow, K. McCann, R. Diswas and R. Van der Wijngaart. 2000. <http://www.nas.nasa.gov/Research/Reports/Techreports/2000/nas-00-009-abstract.html>
- [42] "JavaGenes: Evolving Graphs with Crossover," A. Globus, S. Atsatt, J. Lawton and T. Wipke. 2000. <http://www.nas.nasa.gov/~globus/papers/JavaGenes/paper.html>
- [43] "JavaGenes and Condor: Cycle-Scavenging Genetic Algorithms," A. Globus, E. Langhirt, M. Livny, R. Ramamurthy, M. Solomon and S. Traugott. In *Java Grande 2000*. 2000. <http://www.nas.nasa.gov/~globus/papers/JavaGrande2000/JavaGrandePaper.html>
- [44] "Large-Scale Distributed Computational Fluid Dynamics on the Information Power Grid using Globus," R. Biswas, S. Barnard, S. Saini, R. Van der Wijngaart, M. Yarrow, L. Zechter, I. Foster and O. Larsson. In *7th Symposium on the Frontiers of Massively Parallel Computation*. 1999.
- [45] "Parallel and Distributed Computational Fluid Dynamics: Experimental Results and Challenges," M. Djomehri, R. Biswas, R. Van der Wijngaart and M. Yarrow. In *Proc. of 7th Intl. High Performance Computing Conf.* 2000. Bangalore, India.
- [46] "A Scalable Content-Addressable Network", S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker. ACM SIGGCOM, 2001. <http://www.cs.berkeley.edu/~sylvia/>
- [47] "Condor," M. Livny and e. al. <http://www.cs.wisc.edu/condor>